Day-3-30-11-2023

Game Based Problems-2

Problem-1

Number of ways for playing first move

Two players **A** and **B** are playing **NIM Game** with each other. Both are playing optimally. Player **A** starts the game. The task is to find the number of ways of playing **1st** move for **A** to ensure a winning strategy for **A** if possible, otherwise print **-1**.

NIM Game rules:
Given a number of piles in which each pile contains some numbers of stones/coins. In each turn, a player can choose only one pile and remove any number of stones (at least one) from that pile. The player who cannot move is considered to lose the game (i.e., one who take the last stone is the winner).

Examples:

Input: arr[] = {1, 2, 3}
Output: -1
There is no winning strategy for A no matter how optimally he plays.

Input: arr[] = {2, 4, 5}
Output: 1

Explanation:

In order to play optimally, A will pick one coin from first pile and that's the only optimal move.

- First check who will win the game by taking XOR of all the array elements if XOR is zero then no matter how optimally A play, A will always lose. If XOR is non-zero then go to Step 2.
- We will check for every pile if we can remove some coins from that pile so that after this move, XOR of all the array elements will be zero. So for all piles, one by one we will take xor of all remaining elements of the array and will check if XOR value is greater than the number of coins in the pile. If so, it is not possible to play the first move using this pile because we can only remove coins from the pile in a move, and cannot add coins. Otherwise, we will increment the number of ways.

Sample Input 1:
3 (size of array)
1
2
3

Sample Output 1:
-1

Sample Input 2:
3 (size of array)
2
4
5

Sample Output 2:
1

Solution :

```
n=int(input())
l=[]
res=0
for i in range(n):
    l.append(int(input()))
    res=res^l[i]
if (res==0):
    print("-1")
else:
    cnt=0
    for i in l:
        if i^res<=i:
            cnt=cnt+1
    if cnt==0:
        print("-1")
    else:
        print(cnt)
```

Problem-2

Absolute difference of sum

Predict the winner of the game on the basis of absolute difference of sum by selecting numbers Given an array of N numbers. Two players X and Y play a game where at every step one player selects a number. One number can be selected only once. After all the numbers have been selected, player X wins if the absolute difference between the sum of numbers collected by X and Y is divisible by 4, else Y wins.
Note: Player X starts the game and numbers are selected optimally at every step.

Examples:

Input: a[] = {4, 8, 12, 16}
Output: X

One of the way:
X chooses 4
Y chooses 12
X chooses 8
Y chooses 16
|(4 + 8) – (12 + 16)| = |12 – 28| = 16 which is divisible by 4.
Hence, X wins

Input: a[] = {7, 9, 1}
Output: Y

Explanation:

The following steps can be followed to solve the problem:

Initialize count0, count1, count2 and count3 to 0.
Iterate for every number in the array and increase the above counters accordingly if a[i] % 4 == 0, a[i] % 4 == 1, a[i] % 4 == 2 or a[i] % 4 == 3.
If count0, count1, count2 and count3 are all even numbers then X wins else Y will win.

Sample Input 1:
4 (Size of array)
4
8
12
16

Sample Output 1:
X

Sample Input 2:
3 (Size of array)
7
9
1

Sample Output 2:
Y

Solution :

```
n=int(input())
l=[]
for i in range(n):
  l.append(int(input()))
count=count1=count2=count3=0
for i in l:
  if(i%4==0):
    count=count+1
  elif(i%4==1):
    count1=count1+1
  elif(i%4==2):
    count2=count2+1
  elif(i%4==3):
    count3=count3+1
if(count%2==0 and count1%2==0 and count2%2==0 and count3%2==0):
  print("X")
else:
  print("Y")
```

Problem-3

Game of N stones

Game of N stones where each player can remove 1, 3 or 4
Two players are playing a game with n stones where player 1 always plays first. The two players move in alternating turns and plays optimally. In a single move a player can remove either 1, 3 or 4 stones from the pile of stones. If a player is unable to make a move then that player loses the game. Given the number of stones where n is less than equal to 200, find and print the name of the winner.

Examples:

Input : 4
Output : player 1

Input : 7
Output : player 2

Explanation:

To solve this problem, we need to find each possible value of n as a winning or losing position. Since above game is one of the impartial combinatorial games, therefore the characterization of losing and winning position is valid.

The characteristic properties of winning and losing states are:

All terminal positions are losing positions.
From every winning position, there is atleast one move to a losing position.
From every losing position, every move is to a winning position.
If a player is able to make a move such that the next move is the losing state than the player is at winning state. Find the state of player 1, if player1 is in winning state then player 1 wins the game otherwise player 2 will win.

Consider the following base positions:

Position 0 is the losing state, if the number of stones is 0 than the player1 will unable to make a move therefore player1

loses.

Position 1 is the winning state, if the number of stones is 1 than the player1 will remove the stone and win the game.

Position 2 is the losing state, if the number of stones is 2 than the player1 will remove 1 stone and then player2 will remove the second stone and win the game.

Position 3 is the winning state, if the player is able to take a move such that the next move is the losing state than the player is at winning state, the player1 will remove all the 3 stones

Position 4 is the winning state, if the player is able to take a move such that the next move is the losing state than the player is at winning state, the player1 will remove all the 4 stones

Position 5 is the winning state, if the player is able to take a move such that the next move is the losing state than the player is at winning state, the player1 will remove 3 stones leaving 2 stones, which is the losing state

Position 6 is the winning state, if the player is able to take a move such that the next move is the losing state than the player is at winning state, the player1 will remove 4 stones leaving 2 stones, which is the losing state

Position 7 is the losing state, if the number of stones is 7 than the player1 can remove 1, 3 or 4 stones which all leads to the losing state, therefore player1 will lose.

Sample Input 1:
4

Sample Output 1:
P1

Sample Input 2:
7

Sample Output 2:
P2

Sample Input 3:
201

Sample Output 3:
Invalid Input

```
n=int(input())
if(n<0 or n>200):
  print("Invalid Input")
else:
  if n%7==0 or n%7==2:
    print("P2")
  else:
    print("P1")
```

Problem-4

Piles of boxes

Given N piles containing White(W) and Black(B) boxes only, two players A and B play a game. Player A may remove any number of boxes from the top of the pile having the topmost box White and Player B may remove any number of boxes from the top of the pile having the topmost box Black. If there is a pile with top as W, then player A has remove one or more boxes. Similarly if there is a pile with top as B, then player B has to remove one or more boxes. They play alternating, with the player A first to move. Both players plays optimally.

The task is to find the winner of the game (who cannot make the last move). Player last to move loses the game.

Examples:

Input: N = 2
WBW,BWB
Output: A

Player A can remove all boxes from pile
1. Now player B has to make a choice from pile
2. Whatever choice player B makes, he/she has to make the last move.

Input: N = 3
WWWW,WBWB,WBBW
Output: B

Explanation

The game theory problems, where the player who makes the last move lose the game, are called Misere Nim's Game.

If there are consecutive occurrences of the same box in any pile, we can simply replace them with one copy of the same box, since it is optimal to remove all occurrences of the box present at the top. So, we can compress all piles to get the pile of form BWBWBW or WBWBWB.

Now, suppose there's a pile, which have topmost box differing from its bottom-most box, we can prove that answer remains same even without this pile, because if one player makes a move on this pile, other player makes the next move by removing box from the same pile, resulting in exactly same position for the first player.

If a pile has both top and bottom-most box as same, It requires one or the player to make one extra move. So, just count the number of extra move each player has to make. The player which runs of out extra moves first wins the game.

Input format:

First line of input consists of an integer n, corresponds to the number of piles.

The next n lines of input corresponds to the piles containing White(W) and Black(B) boxes

Output format:

Output consists of a character. Print A if A wins, else print B.

Sample Input 1:
2 (number of piles)
WBW
BWB

Sample Output 1:
A

Sample Input 2 :
3 (number of piles)
WWWW
WBWB
WBBW

Sample Output 2 :
B

```
n=int(input())
l=[]
for i in range(n):
    l.append(input())
ac=0
bc=0
for i in range(n):
    j=len(l[i])
    if (l[i][0]==l[i][j-1]) and ( l[i][0]=="W"):
        ac=ac+1
    if (l[i][0]==l[i][j-1]) and ( l[i][0]=="B"):
        bc=bc+1
if(ac<=bc):
    print("A")
else:
    print("B")
```